Proceedings of the International Conference , “Computational Systems for Health & Sustainability” 17-18, April, 2015 - by R.V.College of Engineering, Bangalore,Karnataka,PIN-560059,INDIA

A Survey On:Cloud Centric Internet of Things and Intelligent Agents

Mahesh B Fattepur
M.Tech CSE – II Sem
R.V College of Engineering Bangalore-560059 India

Jayashree B.Huttanagoudar
M.Tech CSE – II sem
R.V College of Engineering Bangalore-560059, India

Abstract—Cloud Computing and Internet of Things (IoT) are currently two of the most popular Information and Communication Technology paradigms. Intelligent agent is one of the very important concept of artificial intelligent that acts on behalf of users for decision making.

Internet of Things (loT) is a concept that envisions all objects around us as part of internet. loT coverage is very wide and include variety of objects like smart phones, tablets, digital cameras, sensors, etc. Once all these devices are connected with each other, they enable more and more smart processes and services that support our basic needs, economies, environment and health. Such enormous number of devices connected to internet provides many kinds of services and produce huge amount of data and information.

Cloud computing is a model for on-demand access to a shared pool of configurable resources (e.g. compute, networks, servers, storage, applications, services, and software) that can be easily provisioned as Infrastructure (laaS), software and applications (SaaS) . Cloud based platforms help to connect to the things (laaS) around us so that we can access anything at any time and any place in a user friendly manner using customized portals and in built applications (SaaS). Hence, cloud acts as a front end to access Internet of Things.

IoT provides us with lots of sensor data. But the data by themselves do not provide value unless we can turn them into actionable, contextualized information. Big data and data visualization techniques allow us to gain new insights by batch-processing and offline analysis. Real time sensor data analysis and decision making is often done manually but to make it scalable, it is preferably automated. Artificial intelligence provides us the framework and tools to go beyond trivial real-time decision and automation use cases of IoT.

In this paper, we have made survey on the IoT environment, cloud computing, programming languages used to implement intelligent agents and frameworks to integrate them with middleware. Finally, we briefly overview our intelligent agent-oriented, cooperating smart objects and a sensor-cloud infrastructure applications which prove very helpful.

Keywords—Intelligent Agent; Cloud Computing; Internet of Things;Smart Objects;Middleware;

INTRODUCTION

Cloud computing and Internet of Things (IoT) are two popular ICT paradigms that have come to the attention of the research community over the last years. In the Internet of Things (IoT) paradigm many of the objects that surround us will be on the network in one form or another. Radio Frequency Identification (RFID) and sensor network technologies will rise to meet this new challenge, in which

information and communication systems are invisibly embedded in the environment around us. These results in the generation of enormous amounts of data which have to be stored, processed and presented in a seamless, efficient, and easily interpretable form. Cloud computing can provide the virtual infrastructure for such utility computing which integrates monitoring devices, storage devices, analytics tools, visualization platforms and client delivery. The cost based model that Cloud computing offers will enable end-to-end service provisioning for businesses and users to access applications on demand from anywhere.

Smart connectivity with existing networks and context-aware computation using network resources is an indispensable part of IoT. With the growing presence of WiFi and 4G-LTE wireless Internet access, the evolution towards ubiquitous information and communication networks is already evident. However, for the Internet of Things vision to successfully emerge, the computing paradigm will need to go beyond traditional mobile computing scenarios that use smart phones and portables, and evolve into connecting everyday existing objects and embedding intelligence into our environment. The paper is structured as follows. Section II provides look at glance about the IoT and Intelligent Agents. Section III presents the Cloud centric IoT. Section IV overviews the programming languages, IDEs and frameworks which can be used to program the intelligent agents , Section V briefs few recent developments carried out by some leading industries which use the concept of combining cloud centric IoT with AI and finally, in Section VI the conclusion is drawn along with some applications where combination of all the three technologies prove very useful.

LITERATURE SURVEY ON INTERNET OF THINGS AND INTELLIGENT AGENTS

A. INTERNET OF THINGS(DEFINITION ,TRENDS AND ELEMENTS)

Definition:

- According to Cluster of European research projects on IoT[1]: ‘Things’ are active participants in business, information and social processes where they are enabled to interact and communicate among themselves and with the environment by exchanging data and information sensed about the environment, while reacting autonomously to the real/physical world events and influencing it by running processes that trigger actions and create services with or without direct human interaction.

Trends:

Its identified as one of the emerging technologies in IT as noted by Gartner’s Hype Cycle. As measured by the Google search trends there is lot of surf on IoT , WSN and Ubiquitous computing.

IoT Elements:

There are three IoT components which enables seamless ubicomp:

1. Hardware: Sensors and Actuators
2. Middleware: Storage and interpretation through data analysis.
3. Presentation: Visualization and interpretation according to application.

Enabling technologies:

RFID :

- Major breakthrough in embedded communication paradigm which enables design of microchips for wireless data communication.
- Helps in automatic identification through electronic barcode.
- Two types of RFID : Passive RFID(Used in many bank cards) and Active RFID tags(in port containers monitoring cargo).

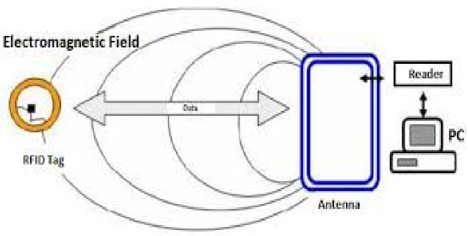

Fig 1. RFID working

Wireless Sensor Networks:

Provide efficient, low cost, low power miniature devices for use in remote sensing applications. Because of above mentioned advantages with WSN we can utilize sensor network of large number of intelligent sensors, enabling the collection, processing, analysis and dissemination of valuable information, gathered in a variety of environments[2].

The components of WSN:

1. WSN hardware: Sensor interfaces, processing units, transceivers units and power supply also they comprise of A/D converters.
2. WSN Communication Stack: Appropriate topology, routing and MAC layer are critical. Nodes communicate with each other and send data to base station either in a single or multiple hops. Stack should act as gateway to the WSN and Internet to communicate to outside world.
3. WSN Middleware: A mechanism to combine cyber infrastructure with a SOA (Service Oriented Architecture) and sensor networks to provide access to heterogeneous sensor resources in a deployment independent manner.

   Platform independent middleware to develop sensor applications is required such as Open Sensor Web Architecture (OSWA)

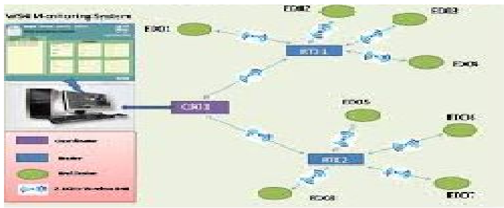

Fig 2. Wireless Sensor Network Working

Secure Data Aggregation:

Required for extending the life time of the network as well as ensuring reliable data collected from sensors.

Addressing Schemes:

- IPV4: Identifies a group in a particular geographical area but not individually.
- Internet mobility attribute in IPV6 may alleviate some identification problem.
- Heterogeneous nature of wireless nodes, variable data types, concurrent operations and confluence of data from devices exacerbates the problem further.
- Address scalability problem due to increased network size.
- It hampers the performance of network, functionality of device and the reliability of data over the network.
- To address these issues URN are used which will create replicas of resources through URL.
- Make use of metadata for data transfer.
- IPV6 allow unique and remote resource access
- WSN and Internet possess different stacks so subnet with a gateway having URN is required.
- The entire network now forms a connectivity from users (high level) to sensors (low level) that is addressable (through URN), accessible (through URL) and controllable (through URC).

Data Storage and analytics:

As large amount of data is gathered from heterogeneous sensor networks storage, ownership and expiry of the data become critical issues. Data have to be stored and used intelligently for smart monitoring and actuation.

It is important to develop artificial intelligence algorithms which could be centralized or distributed based on the need. Novel fusion algorithms need to be developed to make sense of the data collected. State-of-the-art non-linear, temporal machine learning methods based on evolutionary algorithms, genetic algorithms, neural networks, and other artificial intelligence techniques are necessary to achieve automated decision making. Cloud based storage solutions are increasingly becoming popular.

Visualization:

- Critical for an IoT application as this allows the interaction of the user with the environment.
- For a lay person to fully benefit from the IoT revolution, attractive and easy to understand visualization has to be created.
- This will enable policy makers to convert data to knowledge which is critical in fast decision making.
- Data is extracted according to the needs of end user.

Applications of IoT:

The applications can be classified based on the type of network availability, coverage, scale, heterogeneity, repeatability, user involvement and impact. We categorize the applications into four application domains :(1) Personal and Home (2) Enterprise (3) Utilities and (4) Mobile. Personal and Home IoT at the scale of an individual or home, Enterprise IoT at the scale of a community, Utility IoT at a national or regional scale and Mobile IoT which is usually spread across other domains mainly due to the nature of connectivity and scale. There is a huge crossover.

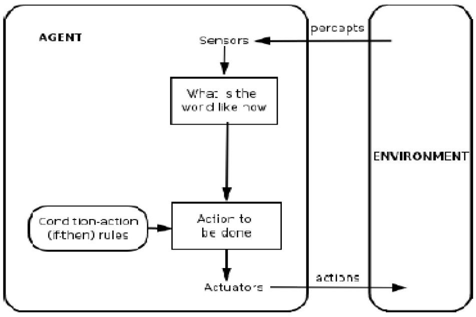

Fig 3. Intelligent Agent

Fig 4. IoT Applications and End Users Based on Data

B. Intelligent Agents

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators. An Intelligent agent is a special software component that can act independent on behalf of its user.

The agent features perfectly fit the[3,4]

- Autonomy: agents/CSOs should be able to perform the majority of their problem solving tasks without the direct involvement of humans or other agents, and they should have a degree of control over their own actions and their own internal state.
- Interaction: agents/CSOs should be able to interact, when they deem appropriate, with other software agents/SOs and humans to complete their own problem solving and support others with their activities where appropriate.
- Responsiveness: agents/CSOs should perceive their environment, in which they are situated and which may be the physical world, a user, a collection of agents, the Internet, and so forth, and respond in a timely fashion to changes that may occur in it.
- Proactiveness: agents/CSOs should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behaviour and take the initiative where and when appropriate.

CONVERGENCE OF IOT WITH CLOUD: SURVEY

The vision of IoT can be seen from two perspectives:

Internet Centric: Internet service is more priority

Thing Centric: Smart objects take center stage

What makes the Cloud-based Internet of Things different than conventional Internet of Things is basically the ability to develop, deploy, run, and manage Things applications online via the Cloud. Fig. 4 and Fig.5 illustrates the main features of the Cloud-based IoT platform and their interaction with the three Cloud computing models of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). also specifies technical solutions to networking Things, interacting Things, and integrating Things with the Cloud

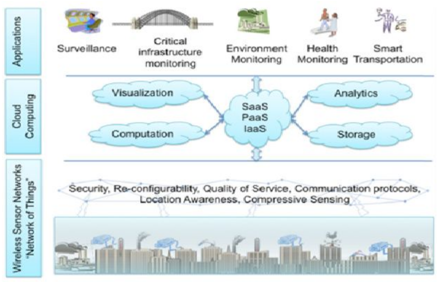

Fig 5.a Conceptual Framework with cloud at the center

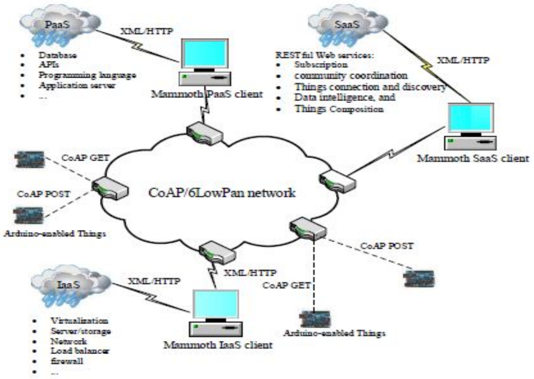

Fig 5.b. the Cloud-based IoT platform

Above two architectures are the online platform that allows

system integrators and solution providers to leverage a complete Things application infrastructure for developing, deploying, operating, and composing Things applications and services that consist of three major modules:

- The CloudThings service platform for Things is a set of Cloud services (IaaS), allowing users to run any applications on Cloud hardware. The CloudThings service platform for Things dramatically simplifies the application development, eliminates need for infrastructure development, shortens time to market, and reduces Things management and maintenance costs. The CloudThings service platform offers users unique device management capabilities. It communicates directly with devices and provides storage to collect Things data and transmit Things events. Vast amount of sensor data can be processed, analyzed, and stored using the computational and storage resources of the Cloud. The CloudThings service platform allows sharing of sensor resources by different users and applications under a flexible usage mode.
- The CloudThings Developer Suite for Things is a set of Cloud service tools (PaaS) for Things application development. These tools include open Web service application programming interfaces (APIs), which provide complete development and deployment capabilities to Things developers.
- he CloudThings Operating Portal for Things is a set of Cloud services (SaaS) that support deployment and handle or support specialized processing services including service subscription management, community coordination, Things connection, Things discovery, data intelligence, and Things composition.

Above modules are realized as follows:

A. Interact with Things Using Constrained Application Protocol (CoAP)

In the CloudThings architecture, we use CoAP[5] to interact with Things. The CoAP is a specialized Web transfer protocol for use with constrained nodes and constrained networks. The protocol is designed for machine-to-machine (M2M) applications such as smart energy and building automation.

CoAP provides a request/response interaction model between application end-points. This supports built-in discovery of services and resources, and includes key concepts of the Web such as URIs and Internet media types. CoAP easily interfaces with HTTP for integration with the Web, while meeting specialized requirements such as multicast support, very low overhead, and simplicity for constrained environments.

Resources are requested and identified by URIs using the Representational State Transfer (REST) methods of GET, PUT, POST and DELETE. In contrast to HTTP, the CoAP exchanges messages asynchronously over UDP (User Datagram Protocol).

B. Networking with Things Using 6LoWPAN

In the CloudThings architecture, we use 6LoWPAN, i.e. IPv6-based Low Power Wireless Area Networks. 6LoWPAN[6] deines message frame formats, fragmentation methods, and header compression techniques required to fit Ipv6/UDP datagrams in the very limited IEEE 802.15.4 frame size.

The 6LoWPAN innovations provide IP access to a wide set of networked devices, which, being low-cost, low-power constrained hosts, could not easily benefit from the huge addressing space of IPv6. 6LoWPAN is able to reduce the IPv6/UDP header while maintaining the main functionalities and the size of the addressing space, thanks to a cross-layer optimization approach.

Routing functionalities are provided by the Routing Protocol for Low power and lossy networks (RPL)[7], which are another IETF (Internet Engineering Task Force) solution discussed in the Routing Over Low power and Lossy networks (ROLL) working group.

C. Integration with the Cloud Using RESTful Web Services

There exist two architectural styles for Web services applications: REST[8] and SOA (Service-Oriented Architecture)[9] . Both of these describe the methods for

designing and developing interoperable services via Web and design principles.. The RESTful protocol is HTTP, which uses HTTP-similar standardized methods (e.g. GET, PUT, POST, DELETE, etc.) to deal with resources. We adopt REST as architectural style, where Things are modeled as RESTful resources and referred as services

OVERVIEW OF PROGRAMMING LANGUAGES , INTEGRATED DEVELOPMENT ENVIRONMENT, PLATFORMS AND FRAMEWORKS FOR INTELLIGENT AGENTS

Most agent programming languages have some underlying platform which implements its semantics. However, agent frameworks exist that are not tightly coupled with one specific programming language. Instead, they are concerned with providing general techniques for relevant aspects such as agent communication and coordination. The most mature languages will be accompanied by some Integrated Development Environment (IDE), intended to enhance the productivity of programmers by automating tedious coding tasks. Typically these will provide functionalities such as project management, creating and editing source files, refactoring, build and run process, and testing.

D. Programming Languages:

Following are the programming languages which are used to program intelligent agents:

1. Declarative Languages: Declarative languages are partially characterised by their strong formal nature, normally grounded on logic.

   Ex: FLUX, Minerva, Dali, and ResPect. Other declarative languages are also grounded on other formalisms, such as CLAIM which finds parts of its roots in the ambient calculus. CLAIM (Computational Language for Autonomous, Intelligent and Mobile Agents[10]) is a high-level declarative agent-oriented programming language.

   It is part of an unified framework called Himalaya[11] (Hierarchical Intelligent Mobile Agents for building Large-scale and Adaptive systems based on Ambients). It combines the main advantages of agent-oriented programming languages, for representing cognitive aspects and reasoning, with those of concurrent languages based on process algebra, for representing concurrency and agent mobility.

2. Imperative Languages: Purely imperative approaches to agent-oriented programming are less common, mainly due to the fact that most abstractions related to agent-oriented design are, typically declarative in nature.

Ex: The JACK Agent Language (JAL) has been developed by a company called Agent Oriented Software.

This is an extension to JAVA In JAL, plans can be composed of reasoning methods and grouped into capabilities which, together, compose a specific ability an agent is supposed to have, thus supporting a good degree of modularisation. Another structuring mechanism present in JAL is the ability to use teams of agents, or agent organisations, a notion that is increasingly important both in agent-oriented design and because of recent developments in self-organising systems.

3. Hybrid Approaches : These are the languages which combine the features of both declarative and imperative languages.

Ex :3APL(An Abstract Agent Programming Language “triple-a-p-l”),Jason ,IMPACT

E. Integrated Development Environments

Integrated Development Environments (IDEs), focus on the programming language level and intend to enhance the productivity by automating tedious coding tasks.

Ex :3APL IDE, Jason IDE, JDE, CAFnE, Visual Soar, Agent-Builder, AgentFactory, and the Living Systems Developer

F. Agent Platforms and Frameworks

These frameworks are more concerned with providing support for aspects such as agent communication and coordination. In this Section we focus on such frameworks.

Ex : TuCSoN, JADE[12], and DESIRE

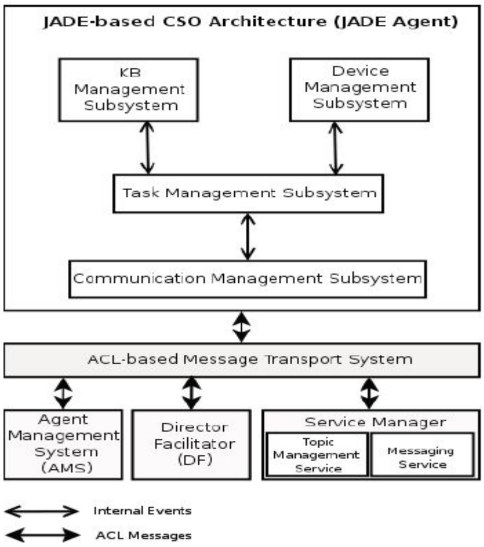

Fig 6. JADE based CSO architecture.

The main CSO architecture components are:

- The Task Management Subsystem, which manages the reactive and proactive tasks of CSOs. In particular,tasks are event-driven and state-based software components encapsulating specific objectives to fulfill through computation, communication, sensing/actuation, and storage management operations. Tasks can be defined as JADE Behaviours or JADEX Plans so their execution is based on the mechanisms provided by the basic JADE behavioral execution model or plan oriented Jadex execution model, respectively.

  Being tasks driven by events, external CSO communication, signals to/from the CSO devices, data to/from the knowledge base (KB) are internally formalized and managed as events.

- The Communication Management Subsystem, which provides a common interface for CSO communications. In particular, message-based communication is based on the ACL-based Messaging Service whereas publish/subscribe coordination is the Topic Management Service. The subsystem is internally organized in handlers. The Communication Manager MessageHandler, which is implemented as Behaviour in JADE and as Plan in Jadex, captures the ACL messages targeting CSOs and translates them into internal events. Moreover, the TCPAdapter and UDP Adapter manage communication with external networked entities based on TCP and UDP, respectively.

- The Device Management Subsystem, which manages the sensing/actuation devices that belong to the CSO. It is organized in a Device Manager handling several Device Adapters. Currently, two Device Adapters are available: the BMF Adapter, which allows to manage WSANs based on BMF, and the SPINE Adapter, which allows to manage BANs based on SPINE. BMF and SPINE are based on IoT standards protocols such as IEEE 802.15.4, ZigBee, and 6LowPan.
- The KB Management Subsystem, which supports CSOs through a knowledge base (KB). It consists of a KBManager, which manages and coordinates different KBAdapters, and a KBAdapter, which manages a KB containing the knowledge of the CSO. KB can be local and/or remote and archives information that can be shared among tasks.

DEVELOPMENTS IN INDUSTRY WHICH ARE PREFERING CLOUD CENTRIC IOT AND INTELLIGENT AGENTS

As consumers get used to buying so-called smart devices, they are eventually going to expect them to actually be smart. They might even expect them to be smart all the time.[13]

1. The Nvidia new mobile GPU , called the Tegra X1 , which is the foundation of the company’s new DRIVE PX automotive computer. Nvidia claims that it will allow cars to spot available parking spaces , park themselves and pick up drivers like a valet, and be able to distinguish between the various types of vehicles a car might encounter while on the road.
2. IBM has built a low power neurosynaptic(i.e, modeled after the brain) chip called SyNAPSE , which is designed for machine learning tasks such as object recognition and consumes less power.
3. Qualcomm has built Zeroth which will be embedded within the next generation of devices.
4. TeraDeep, a startup is working on deep learning algorithms that can turn on traditional ARM and other mobile processor platforms. It says that deep learning modules can be directly embedded into other connected devices, as well.

Cloud computing plays a big role for consumers as AI makes it a way further into the internet of things. The cloud will still process the data for applications that analyze aggregate user data.

CONCLUSION

In this paper we have made a survey on Internet of Things which is having cooperating smart objects in its environment , and the cloud infrastructure that connects, coordinates and facilitates to communicate with each other through its services such as SaaS, PaaS and IaaS. Finally the survey includes a brief information about the programming languages , IDEs and agent platforms and frameworks which are essential to embed the intelligence in the middleware that could be used to integrate IoT with Cloud so that the applications that are running in cooperating smart object environment become more intelligent enough to act on behalf of human beings. The recent developments in leading industries area appreciable.

ACKNOWLEDGMENT

The authors are grateful to Dr.S.Sridhar, Professor and Dean-Cognitive and Central Computing, of our college for the guidance to prepare this review article and also thankful to Dr.G.Shobha, HoD/CSE and Dr.N.K.Srinath, PG Dean (CSE) for the support offered.

REFERENCES

[1]. H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, Vision and challenges for realising the Internet of Things, Cluster of European Research Projects on the Internet of Things—CERP IoT, 2010.

[2]. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[3]. M. Vinyals, J. A. Rodriguez-Aguilar, J. Cerquides, “A Survey on Sensor Networks from a Multiagent Perspective,” The Computer Journal, 54(3), pp. 455-470, 2010.

[4]. A. Rogers, D. Corkill, and N.R. Jennings, N. R. “Agent technologies forsensor networks,” IEEE Intelligent Systems, 24, pp. 13-17, 2009.

[5]. Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "Constrained Application Protocol (CoAP) draft-ietf-core-coap-1",

http://datatracker.ietf.org/doc/draft-ietf-core-coap/, retrieved by 20 July

[6]. Zach Shelby and Carsten Bormann, 6LoWPAN: The Wireless Embedded Internet. Wiley, 2009..

[7]. T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis,K. Pister, R. Struik, and J. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", IETF Internet Draft draft-ietf-roll-rpl-19,2011, http://tools.ietf.org/html/draft-ietf-roll-rpl-19, retrieved by 20 July 2012..

[8]. R. Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Ph.D. dissertation, University of California, Irvine, 2000.

[9]. Thomas Erl, Service-Oriented Architecture (SOA): Concepts,Technology, and Design. Prentice Hall, 2005.

[10]. A. El Fallah Seghrouchni and A. Suna. CLAIM: A computational language for autonomous, intelligent and mobile agents. In M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors, Programming Multiagent

Systems, first international workshop (Pro- MAS'03), volume 3067 of LNCS, pages 90–110.Springer Verlag, 2004.

[11]. A. El Fallah Seghrouchni and A. Suna. Himalaya Framework: Hierarchical Intelligent Mobile Agents for building Large-scale and Adaptive sYstems based on Ambients. In T. Ishida, L. Gasser, and H. Nakashima, editors, Proceedings of Massive Multi-Agent Systems workshop, number 3446 in LNAI, pages 202–216. Springer Verlag, 2005..

[12]. F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi. JADE — a java agent development framework. In Bordini et al. [5], chapter 5, pages 125–148.

[13]. https://gigaom.com/2015/01/06/ai-is-coming-to-iot-and-not-all-the-brains-will-be-in-the-cloud/